Accepted for IEEE Transactions on Circuits and Systems II Express Briefs, May 2010. This draft will not be updated any more. Please refer to IEEE Xplore for the final version.
*) The final publication will appear with the modified title *"A Scalable VLSI Architecture for Soft-Input Soft-Output Single Tree-Search Sphere Decoding"*.

A Scalable VLSI Architecture for Soft-Input Soft-Output Depth-First Sphere Decoding*

Ernst Martin Witte, Filippo Borlenghi, Gerd Ascheid, *Senior IEEE*, Rainer Leupers, Heinrich Meyr, *Fellow IEEE*

***Abstract*—Multiple-input multiple-output (MIMO) wireless transmission imposes huge challenges on the design of efficient hardware architectures for iterative receivers. A major challenge is soft-input soft-output (SISO) MIMO demapping, often approached by sphere decoding (SD). In this paper, we introduce the—to our best knowledge—first VLSI architecture for SISO SD applying a single tree-search approach. Compared with a soft-output-only base architecture similar to the one proposed by Studer et al. in IEEE J-SAC 2008, the architectural modifications for soft input still allow a one-node-per-cycle execution. For a 4×4 16-QAM system, the area increases by 57 % and the operating frequency degrades by 34 % only.**

***Index Terms*—VLSI architecture, Schnorr-Euchner (SE) enumeration, iterative multiple-input multiple-output (MIMO) decoding, soft-input soft-output (SISO) sphere decoding (SD)**

I. Introduction

Multiple-input multiple-output (MIMO) wireless transmissions utilizing spatial multiplexing achieve an increased spectral efficiency compared with single-antenna systems. This improvement comes at the cost of an increased signal-demapping complexity, which becomes particularly critical for iterative receivers [1]. Recent developments of soft-input soft-output (SISO) MIMO-demapping algorithms reduced this complexity significantly. Prominent demapping algorithms are k-best and list-based approaches [2], [3], Markov chain Monte Carlo algorithms (MCMC) [4] and single tree-search (STS) sphere decoders (SD) [5]. The STS approach is often preferred since it guarantees max-log maximum a posteriori (MAP) optimality.

Efficient VLSI implementations have been proposed for soft-output-only STS SDs [6], [7] exploiting geometric properties of QAM constellations. These geometric relations help determining a search order, defined as *enumeration*, leading to a fast average tree-search convergence. The SISO STS complexity has been prohibitive for VLSI implementations so far, because geometric relations are not applicable directly. Recent improvements of soft-input enumeration strategies moved SISO STS SD closer to VLSI architectures [8].

Contributions: In this paper, we introduce the—to our best knowledge—first VLSI architecture for SISO STS SD. It is based on a soft-output-only architecture following the one-node-per-cycle (ONPC) paradigm used by [6]. The SISO modifications are modular enough to be applied to other existing STS SD architectures and still allow ONPC execution. Compared with a soft-output-only architecture, the area increases by 57 % and the clock frequency degrades by 34 % for a 4×4 16-QAM system. Thus, this architecture enables STS-based iterative wireless MIMO receivers.

The paper is organized as follows: Section II sums up the basics of SISO STS SD, extended by the soft-input enumeration strategy in Section III. Section IV describes important implementation aspects of the scalable VLSI architecture. In Section V the parameter design space of the SISO STS architecture as well as area, timing and throughput are discussed.

Manuscript received October 26, 2009; revised April 5, 2010. This work has been supported by the UMIC (Ultra High-Speed Mobile Information and Communication) Research Centre at the RWTH-Aachen University.

The authors are with the Institute for Integrated Signal Processing Systems, RWTH-Aachen University, D-52056 Aachen, Germany (email: {witte,borlenghi,ascheid,leupers,meyr}@iss.rwth-aachen.de).

II. Single Tree-Search Soft-Input Sphere Decoding

A spatial-multiplexing MIMO scheme with M_{T} transmit and $M_{\mathrm{R}} \geq M_{\mathrm{T}}$ receive antennas is assumed [1]. Each transmit antenna sends one of the 2^Q complex elements of the symbol set $\mathcal{O}$ defined by the modulation alphabet, which is assumed to be the same for every antenna. Each vector $\mathbf{s} = [s_1, ..., s_{M_{\mathrm{T}}}]^T \in \mathcal{O}^{M_{\mathrm{T}}}$ results from mapping $M_{\mathrm{T}}Q$ bits $x_{i,b} \in \{+1, -1\}$ to an element of $\mathcal{O}^{M_{\mathrm{T}}}$, with i being the antenna index and b the bit index for one scalar symbol s_i.

The received symbol vector $\mathbf{y} \in \mathbb{C}^{M_{\mathrm{R}}}$ is given by $\mathbf{y} = \mathbf{Hs} + \mathbf{n}$, where $\mathbf{H} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{T}}}$ is the channel matrix and $\mathbf{n} \in \mathbb{C}^{M_{\mathrm{R}}}$ is a white circular Gaussian noise vector with variance N_0 per element. For tree-search SD, $\mathbf{H}$ is typically QR-decomposed (QRD) with $\mathbf{H} = \mathbf{QR}$, $\mathbf{Q} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{T}}}$ and $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$ and $\mathbf{R} \in \mathbb{C}^{M_{\mathrm{T}} \times M_{\mathrm{T}}}$ being an upper triangular matrix [1], [5]. With $\tilde{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$ and $\tilde{\mathbf{n}} = \mathbf{Q}^H\mathbf{n}$, this results in

$$\tilde{\mathbf{y}} = \mathbf{Rs} + \tilde{\mathbf{n}} \quad . \tag{1}$$

According to [5], the triangular matrix $\mathbf{R}$ in equation (1) allows to formulate the SISO max-log MAP MIMO detection problem as STS within a 2^Q-ary complete tree. The tree levels correspond to the M_{T} antennas, each node $s_i \in \mathcal{O}$ on tree level i is a received symbol candidate, with s_1 being a leaf node. An exhaustive search in such a tree leads to a worst-case run-time complexity of $O(2^{QM_{\mathrm{T}}})$. As formalized in equations (2) to (4), metric increments $\mathcal{M}_{\mathrm{C}}(s_i)$ for channel-based and $\mathcal{M}_{\mathrm{A}}(s_i)$ for a priori-based information are summed up to a total increment $\mathcal{M}_{\mathrm{P}}(s_i)$. $\mathrm{P}[s_i]$ is the symbol probability computed from the a priori log-likelihood ratios (LLRs) $L^{\mathrm{A}}_{i,b}$.

$$\mathcal{M}_{\mathrm{A}}(s_i) = -\log \mathrm{P}[s_i] \tag{2}$$

$$\mathcal{M}_{\mathrm{C}}(s_i) = \frac{1}{N_0}|\tilde{y_i} - \sum_{j=i}^{M_{\mathrm{T}}} R_{i,j}s_j|^2 \tag{3}$$

$$\mathcal{M}_{\mathrm{P}}(s_i) = \mathcal{M}_{\mathrm{C}}(s_i) + \mathcal{M}_{\mathrm{A}}(s_i) \tag{4}$$

The sum of metric increments along a path from the root to node s_i yields the partial metric $\mathcal{M}_\mathrm{P}(\mathbf{s}^{(i)})$ for a partial symbol vector $\mathbf{s}^{(i)} = [s_i, ..., s_{M_\mathrm{T}}]^T$:

$$\mathcal{M}_\mathrm{P}(\mathbf{s}^{(i)}) \;=\; \sum_{j=i}^{M_\mathrm{T}} \mathcal{M}_\mathrm{P}(s_j) \qquad (5)$$

During a STS, the MAP solution $\mathbf{s}^{\mathrm{MAP}}$, its bits $x_{i,b}^{\mathrm{MAP}}$ and metric $\lambda^{\mathrm{MAP}} = \mathcal{M}_\mathrm{P}(\mathbf{s}^{\mathrm{MAP}})$ and extrinsic counter-hypothesis metrics $\Lambda_{i,b}^{\overline{\mathrm{MAP}}}$ are computed by successively improving the current metrics $\lambda^{\mathrm{MAP,cur}}$ and $\Lambda_{i,b}^{\overline{\mathrm{MAP}},\mathrm{cur}}$. $L_{i,b}^\mathrm{E}$ are extrinsic LLRs with

$$\begin{aligned}
\mathbf{s}^{\mathrm{MAP}} &= \underset{\mathbf{s}\in\mathcal{O}^{M_\mathrm{T}}}{\arg\min}\,\{\mathcal{M}_\mathrm{P}(\mathbf{s})\} \\
\Lambda_{i,b}^{\overline{\mathrm{MAP}}} &= \min_{\mathbf{s}\in\mathcal{O}^{M_\mathrm{T}}\wedge x_{i,b}\neq x_{i,b}^{\mathrm{MAP}}} \{\mathcal{M}_\mathrm{P}(\mathbf{s})\} - L_{i,b}^\mathrm{A} x_{i,b}^{\mathrm{MAP}} \\
L_{i,b}^\mathrm{E} &= \left(\Lambda_{i,b}^{\overline{\mathrm{MAP}}} - \lambda^{\mathrm{MAP}}\right) x_{i,b}^{\mathrm{MAP}} \quad .
\end{aligned}$$

These metric computations dominate the detection complexity. For a depth-first tree search, the pruning of sub-trees lying outside a hypersphere with a radius not improving $\lambda_{i,b}^{\overline{\mathrm{MAP}},\mathrm{cur}} = \Lambda_{i,b}^{\overline{\mathrm{MAP}},\mathrm{cur}} + L_{i,b}^\mathrm{A} x_{i,b}^{\mathrm{MAP}}$ provides a heuristic for complexity reduction which is sensitive to the visiting order $[s_i^{(1)}, ..., s_i^{(|\mathcal{O}|)}]$. A Schnorr-Euchner (SE) order [9] provides a very fast search convergence by the following pruning criteria [6], typically defining the pruning metrics $\mathcal{M}_{\mathrm{prn},j}^{\mathrm{down}} := \mathcal{M}_{\mathrm{prn},j}^{\mathrm{sibl.}} := \mathcal{M}_\mathrm{P}(\mathbf{s}^{(j)})$:

$$\mathcal{M}_{\mathrm{prn},j}^{\mathrm{down}} \geq \max\left\{ \lambda_{i,b}^{\overline{\mathrm{MAP}},\mathrm{cur}} \,\middle|\, i < j \vee x_{i,b} \neq x_{i,b}^{\mathrm{MAP,cur}}, \forall b \right\} \qquad (6)$$

$$\mathcal{M}_{\mathrm{prn},j}^{\mathrm{sibl.}} \geq \max\left\{ \lambda_{i,b}^{\overline{\mathrm{MAP}},\mathrm{cur}} \,\middle|\, i \leq j \vee x_{i,b} \neq x_{i,b}^{\mathrm{MAP,cur}}, \forall b \right\} \qquad (7)$$

If inequality (6) holds, the current node and its sub-tree are pruned, otherwise a step down is performed in the tree. If inequality (7) holds, the enumeration on level j stops, otherwise the sibling of the current node is enumerated. The arguments of the $\max$ operators in (6) and (7) are the sets $\mathcal{A}$ and $\mathcal{B}$ respectively in [6]. We define an *examined node* (as used in [6] and [7]) as a node s_j that has been checked against at least one pruning criterion, leading to the complexity measure *number of examined nodes per detected symbol vector* N_en.

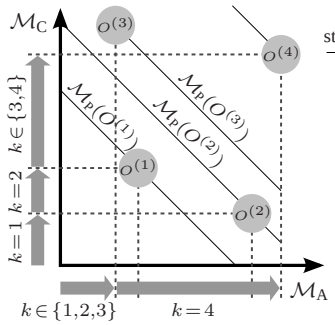
If a leaf node with $\mathcal{M}_\mathrm{P}(\mathbf{s}) \geq \lambda^{\mathrm{MAP,cur}}$ is not pruned by inequalities (6) or (7), the values $\{\Lambda_{i,b}^{\overline{\mathrm{MAP}},\mathrm{cur}} | x_{i,b} \neq x_{i,b}^{\mathrm{MAP,cur}}\}$ need to be updated by $\min\{\Lambda_{i,b}^{\overline{\mathrm{MAP}},\mathrm{cur}}, \mathcal{M}_\mathrm{P}(\mathbf{s}) - L_{i,b}^\mathrm{A} x_{i,b}^{\mathrm{MAP,cur}}\}$. Otherwise, if $\mathcal{M}_\mathrm{P}(\mathbf{s}) < \lambda^{\mathrm{MAP,cur}}$, the current leaf becomes the new MAP solution and the extrinsic counter-hypothesis metrics $\{\Lambda_{i,b}^{\overline{\mathrm{MAP}},\mathrm{cur}} | x_{i,b}^{\mathrm{MAP,old}} \neq x_{i,b}^{\mathrm{MAP,cur}}\}$ are updated by $\min\{\Lambda_{i,b}^{\overline{\mathrm{MAP}},\mathrm{cur}}, \lambda^{\mathrm{MAP,old}} - L_{i,b}^\mathrm{A} x_{i,b}^{\mathrm{MAP,cur}}\}$.

Many methods exist to reduce N_en, like sorted QRD (SQRD) [10] and extrinsic LLR clipping [5]. The latter one limits the allowed range for $L_{i,b}^\mathrm{E}$ to $|L_{i,b,\mathrm{clipped}}^\mathrm{E}| \leq L_\mathrm{max}^\mathrm{E}$, which leads to clipped extrinsic metrics $\Lambda_{i,b,\mathrm{clipped}}^{\overline{\mathrm{MAP}}}$:

$$\Lambda_{i,b,\mathrm{clipped}}^{\overline{\mathrm{MAP}}} = \max\left\{\lambda^{\mathrm{MAP}} - L_\mathrm{max}^\mathrm{E}, \min\left\{\lambda^{\mathrm{MAP}} + L_\mathrm{max}^\mathrm{E}, \Lambda_{i,b}^{\overline{\mathrm{MAP}}}\right\}\right\} \qquad (8)$$

Please note that equation (8) is stricter than the $\min\{\}$ function used in [5] where a post-processing step is used to guarantee $|L_{i,b,\mathrm{clipped}}^\mathrm{E}| \leq L_\mathrm{max}^\mathrm{E}$ for proper channel decoding. In [5], this saves 50 % of the comparisons required for clipping. Experiments indicate that $\mathbb{E}[N_\mathrm{en}]$ differs only marginally between the two clipping methods. Moreover, radius tightening further reduces N_en. A hardware-friendly approximation of $\mathcal{M}_\mathrm{A}(s_i)$ for statistically independent symbols, including tightening and still guaranteeing max-log-optimal a posteriori LLRs, has been proposed in [5] (with unipolar bits $d_{i,b} = \frac{1}{2}(1 - x_{i,b} \cdot \mathrm{sign}(L_{i,b}^\mathrm{A}))$):

$$\mathcal{M}_\mathrm{A}(s_i) = -\log \mathrm{P}[s_i] \approx \sum_{b=1}^{Q} \begin{cases} |L_{i,b}^\mathrm{A}|, & d_{i,b} = 1 \\ 0, & \text{otherwise} \end{cases} \qquad (9)$$



step k	$\mathcal{M}_\mathrm{P}(s_{\mathrm{A},i}^{(k)})$	$\mathcal{M}_\mathrm{P}(s_{\mathrm{C},i}^{(k)})$	$s_i^{(k)}$
1	$\mathcal{M}_\mathrm{P}(O^{(2)}) < \mathcal{M}_\mathrm{P}(O^{(3)})$		$O^{(2)}$
2	$\mathcal{M}_\mathrm{P}(O^{(1)}) < \mathcal{M}_\mathrm{P}(O^{(3)})$		$O^{(1)}$
3	$\mathcal{M}_\mathrm{P}(O^{(4)}) > \mathcal{M}_\mathrm{P}(O^{(3)})$		$O^{(3)}$
	$\mathcal{M}_\mathrm{P}(O^{(4)}) > \mathcal{M}_\mathrm{P}(O^{(1)})$		*skipped*
	$\mathcal{M}_\mathrm{P}(O^{(4)}) > \mathcal{M}_\mathrm{P}(O^{(2)})$		*skipped*
4	$\mathcal{M}_\mathrm{P}(O^{(4)}) = \mathcal{M}_\mathrm{P}(O^{(4)})$		$O^{(4)}$

Fig. 1. Hybrid-enumeration example, k^{th} symbol in SE order: $O^{(k)} \in \mathcal{O}$.

III. The Hybrid-Enumeration Algorithm

A major issue of SD algorithms is the *enumeration* process, namely the determination of the SE order $[s_i^{(1)}, ..., s_i^{(|\mathcal{O}|)}]$ on a level i with $s_i^{(k)}$ representing the k^{th} candidate for node s_i, in ascending order of $\mathcal{M}_\mathrm{P}$. A straightforward implementation by computing and fully sorting the set $\{\mathcal{M}_\mathrm{P}(s_i^{(k)})\}$ is very expensive and inefficient. For the soft-output-only case, the geometric properties of the QAM constellation can be exploited to avoid full sorting and thus save most of the computations, as proposed in [6], [7], [11]. However, in iterative receivers these optimizations are not usable directly because the geometry-based order is scrambled by the a priori information. A viable approach towards efficient soft-input enumeration is given by the *hybrid-enumeration* algorithm presented in [8]. Its basic idea is to split the enumeration of $\{\mathcal{M}_\mathrm{P}(s_i^{(k)})\}$ into two concurrent enumerations of $\{\mathcal{M}_\mathrm{C}(s_i^{(k)})\}$ and $\{\mathcal{M}_\mathrm{A}(s_i^{(k)})\}$.

On the one hand, the enumeration of $\{\mathcal{M}_\mathrm{C}(s_i^{(k)})\}$ is the same as in the soft-output-only case, thus allowing to reuse any of the related aforementioned efficient methods, even in later iterations. On the other hand, the enumeration of $\{\mathcal{M}_\mathrm{A}(s_i^{(k)})\}$ is efficient as well since the linear sorting of the symbol set $\mathcal{O}$ needs to be performed independently only once per antenna.

According to [8], the channel- and a priori-based enumerations independently select candidate symbols $s_{\mathrm{C},i}^{(k)}$ and $s_{\mathrm{A},i}^{(k)}$ at each step k. The hybrid enumeration simply selects the candidate with the lower metric $\mathcal{M}_\mathrm{P}$ between these two.

As visualized in Figure 1, the strict SE order is not preserved, hence the inequality $\mathcal{M}_\mathrm{P}(s_i^{(k)}) \leq \mathcal{M}_\mathrm{P}(s_i^{(l)}), \forall l > k$ does not hold any more. Thus, a modification of the pruning criteria is needed to avoid the erroneous exclusion of the MAP or counter-hypothesis solutions. For $l > k$, the inequalities $\mathcal{M}_\mathrm{C}(s_{\mathrm{C},i}^{(k)}) \leq \mathcal{M}_\mathrm{C}(s_{\mathrm{C},i}^{(l)})$ and $\mathcal{M}_\mathrm{A}(s_{\mathrm{A},i}^{(k)}) \leq \mathcal{M}_\mathrm{A}(s_{\mathrm{A},i}^{(l)})$ lead to

$\mathcal{M}_{\mathrm{C}}(s_{\mathrm{C},i}^{(k)}) + \mathcal{M}_{\mathrm{A}}(s_{\mathrm{A},i}^{(k)}) \leq \mathcal{M}_{\mathrm{P}}(s_i^{(l)})$, providing an alternative lower bound for tree pruning. Thus, in [8] the pruning metric of inequality (7) on the current tree level i is re-defined as

$$\mathcal{M}_{\mathrm{prn},i}^{\mathrm{sibl.}} := \mathcal{M}_{\mathrm{C}}(s_{\mathrm{C},i}^{(k)}) + \mathcal{M}_{\mathrm{A}}(s_{\mathrm{A},i}^{(k)}) + \mathcal{M}_{\mathrm{P}}(\mathbf{s}^{(i+1)}) \quad . \quad (10)$$

Compared with the SE order, pruning metric (10) preserves the error-rate performance at the price of a slight increase in N_{en}. For a more detailed description and analysis of the hybrid-enumeration algorithm, the reader is referred to [8].

IV. A VLSI Architecture for STS Soft-Input Sphere Decoding

In this section, a VLSI architecture for SISO STS SD is introduced. It is derived from a soft-output-only depth-first STS base architecture extended by soft-input processing. The main challenges are discussed that arise from the implementation of efficient soft-input extensions according to the hybrid-enumeration scheme. Further algorithmic optimizations such as LLR correction proposed in [5] are orthogonal to the base architecture and can be implemented on top of it.

A. Soft-Output-Only Base Architecture

The soft-output-only base STS architecture, composed of the light gray blocks in Figure 2, follows the ONPC execution principle used by Studer et al. in [6]. Its architectural structure is derived from the observation that the tree search is composed of three basic control-flow steps:

i) Vertical steps (①) down from tree level i to $i-1$ enumerate the first child node $s_{i-1}^{(1)}$ of a parent node $s_i^{(k)}$. This requires a quantization step $\mathcal{Q}$ to find the QAM symbol next to $\tilde{y}_i$, followed by the computation of $\mathcal{M}_{\mathrm{P}}(s_{i-1}^{(1)})$. The result of $\mathcal{Q}$ is used to initialize the enumeration on the tree level $i-1$ and by the pruning-criteria check for $s_{i-1}^{(1)}$.

ii) Horizontal steps (②) on a tree level i enumerate the node $s_i^{(k+1)}$ after enumerating the node $s_i^{(k)}$ and its sub-tree. This category also includes steps back from a child node s_{i-1} to the next sibling $s_i^{(k+1)}$ of its parent node $s_i^{(k)}$.

iii) Pruning-criteria checks (③) for a node $s_i^{(k)}$ determine if either a *vertical step* to the child $s_{i-1}^{(1)}$, a *horizontal step* to the sibling $s_i^{(k+1)}$ or a *horizontal step* to its parent's sibling $s_{i+1}^{(l+1)}$ has to be performed next. The $\mathcal{M}_{\mathrm{P}}$ *history* (④) unit stores the partial metrics $\mathcal{M}_{\mathrm{P}}(\mathbf{s}^{(i)})$, recursively implements equation (5) and provides its result to unit ③ for pruning and LLR clipping by equation (8).

In a depth-first SD, the tree-traversal control flow exhibits severe data and control dependencies. In order to achieve a throughput of one examined node per cycle, the base architecture executes the pruning check for node $s_i^{(k)}$ concurrently with the steps towards $s_{i-1}^{(1)}$ and $s_i^{(k+1)}$ in cycle n. If the pruning check selects $s_{i-1}^{(1)}$, $s_i^{(k+1)}$ is saved in a *preferred-siblings cache* (⑤) for later use during a step up in the tree. Thus, in cycle $n+1$ the availability of a valid node for the next pruning check is guaranteed.

The enumeration unit of the base architecture employs the column-wise zig-zag enumeration strategy (⑥) presented in [11]. Compared with circular PSK-like enumeration [6], the column-wise enumeration allows a much more regular hardware implementation. Furthermore, for 64 QAM and higher modulation orders it requires less comparisons.

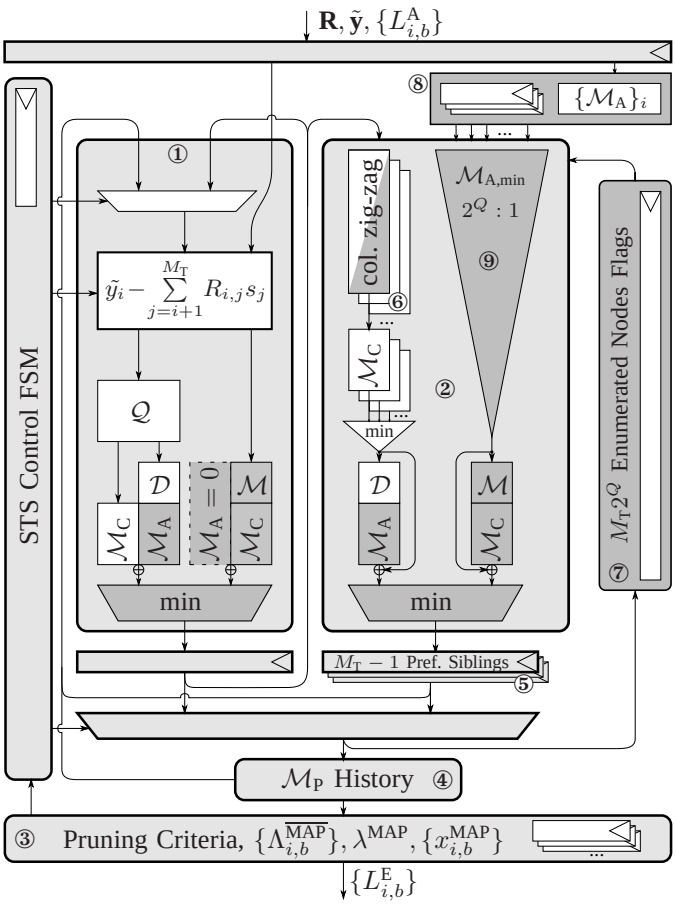


Fig. 2. Block diagram of the proposed soft-input STS SD VLSI architecture. Units added/modified for soft-input are emphasized by dark gray background. Legend: Mapper $\mathcal{M}$, Demapper $\mathcal{D}$, Quantizer $\mathcal{Q}$.

Since there is no assumption on the mapping between QAM symbols and bits, two run-time-programmable lookup tables, named mapper $\mathcal{M}$ and demapper $\mathcal{D}$ respectively, are used for the conversion between the symbol and the bit representations.

B. Soft-Input Extensions

In order to extend the base architecture presented in Section IV-A, mainly extra units for the a priori-based enumeration have to be added, along with slight changes in the column-wise zig-zag implementation. These extensions correspond to the dark gray units in Figure 2.

1) Enumerated-nodes flags: Both channel- and a priori-based enumeration units have to skip nodes that have already been enumerated, because the local enumeration orders for $\mathcal{M}_{\mathrm{C}}$ and $\mathcal{M}_{\mathrm{A}}$ differ from the global enumeration order. Therefore, both units need the list of enumerated nodes to guarantee that each node is enumerated only once. This flag vector of 2^Q bits per antenna is maintained in unit ⑦.

2) Modified column-wise zig-zag enumeration: Skipping an arbitrary number of nodes implies modifications to the column-wise zig-zag implementation (⑥). Compared with the

base architecture, the new column-enumeration unit does not keep internal zig-zag states any more. Instead, each column enumeration performs a minimum search over the linear distances between the quantized imaginary part $\mathcal{Q}(\mathrm{Im}\{\tilde{y}_i - \sum_{j=i+1}^{M_\mathrm{T}} R_{i,j}s_j\})$ and all rows $\{\mathrm{Im}\{s_i|s_i \in \mathcal{O}\}\}$ masked by the enumerated-nodes flags. The hardware complexity increases only moderately, because distance computations are the same for all columns and operate on words of only $Q/2+1$ bits.

3) A priori-based enumeration: With d_i being the decimal representation of the bit vector $[d_{i,Q}, ..., d_{i,1}]$, a mapping of d_i to the corresponding symbol $s_i(d_i)$, $\mathcal{M}_\mathrm{A}(d_i) = \mathcal{M}_\mathrm{A}(s_i(d_i))$ and an order defined by $s_i(d_i^{(k)}) = s_{\mathrm{A},i}^{(k)}$, one problem of enumerating $\{\mathcal{M}_\mathrm{A}\}_i = \{\mathcal{M}_\mathrm{A}(d_i)|0 \le d_i < 2^Q\}$ is the lack of relations among a priori LLRs. Thus, the only known solution is the full computation and sorting of $\{\mathcal{M}_\mathrm{A}\}_i$.

First, the computation of $\{\mathcal{M}_\mathrm{A}\}_i$ (⑧) requires $2^Q - Q - 1$ additions per antenna and received vector. Due to the ONPC principle and the structure of (9), the number of hardware adders can be reduced by resource sharing. The first enumeration step always results in $d_i^{(1)} = 0$ and $\mathcal{M}_\mathrm{A}(d_i^{(1)}) = 0$, thus the subset $\{\mathcal{M}_\mathrm{A}\}_{i,\mathrm{L}} = \{\mathcal{M}_\mathrm{A}(d_i)|1 \le d_i \le 2^{Q-1}\}$ can be computed concurrently. In the second step, $\mathcal{M}_\mathrm{A}(d_i^{(2)}) = \min_{\forall b}|L_{i,b}^\mathrm{A}|$ can be enumerated since $\mathcal{M}_\mathrm{A}(d_i^{(2)}) \in \{\mathcal{M}_\mathrm{A}\}_{i,\mathrm{L}}$, while the subset $\{\mathcal{M}_\mathrm{A}\}_{i,\mathrm{H}} = \{\mathcal{M}_\mathrm{A}(d_i)|2^{Q-1} < d_i < 2^Q\}$ can be computed. This approach only requires $2^{Q-1} - 1$ adders independently from M_T, yielding adder savings of 36 % for 16 QAM and 45 % for 64 QAM. Furthermore, for an ONPC architecture, no latency is added since the subsets $\{\mathcal{M}_\mathrm{A}\}_{i,\mathrm{L}}$ and $\{\mathcal{M}_\mathrm{A}\}_{i,\mathrm{H}}$ can be computed during the enumeration of $s_{\mathrm{A},i}^{(1)}$ and $s_{\mathrm{A},i}^{(2)}$. Further resource sharing would result in limited gains while significantly increasing irregularity.

The second issue is sorting $\{\mathcal{M}_\mathrm{A}\}_i$. Since latency is typically a serious issue for run-time constrained depth-first SD, an approach has been chosen that does not add latency for the sorting of $\{\mathcal{M}_\mathrm{A}\}_i$. The ONPC principle allows a minimum search (⑨) for $\mathcal{M}_{\mathrm{A,min}}$ over the set $\{\mathcal{M}_\mathrm{A}\}_i$ for the enumeration of the current antenna i, masked by the enumerated-nodes flags. The resulting binary tree of compare-select (CS) units would dominate the critical path already for 16 QAM.

However, the properties of equation (9) can be exploited to remove almost all comparators and CS dependencies for the first three CS levels. The principle can be explained easily by considering the removal of the first level: for pairs of $\{\mathcal{M}_\mathrm{A}(s_i^{(k)}), \mathcal{M}_\mathrm{A}(s_i^{(l)})\}$ with only one bit $\{b|x_{i,b}^{(k)} \neq x_{i,b}^{(l)}\}$ the larger metric $\mathcal{M}_\mathrm{A}(s_i^{(\{k,l\})})$ is the one with $x_{i,b}^{(\{k,l\})} \neq \mathrm{sign}(L_{i,b}^\mathrm{A})$. This kind of decision does not need any metric comparison but can be determined by single-bit comparisons of sign bits and enumerated-nodes flags. Selecting the minimum of 4-tuples (first two CS tree levels) differing in only two bits $\{b_{\{m,n\}}|x_{i,b_{\{m,n\}}}^{(k)} \neq x_{i,b_{\{m,n\}}}^{(l)}\}$ requires an additional comparison $|L_{i,b_m}^\mathrm{A}| \gtrless |L_{i,b_n}^\mathrm{A}|$. However, this extra comparison is the same for all 4-tuple sub-trees and does not depend on intermediate results generated in the CS tree. Therefore, the critical path is significantly reduced. The extension to 8-tuples (first three CS tree levels) has a total of only six parallel comparators. Thus, only one CS unit and two 8:1 multiplexers are required for 16 QAM and only seven CS units and eight 8:1 multiplexers for 64 QAM. Compared with a full CS tree, the comparator savings are 53 % in total and 50 % in the critical path for 16 QAM and 79 % in total and 33 % in the critical path for 64 QAM. Extensions to higher orders than 8-tuples are possible but would result in an exponential complexity increase.

4) Pruning-criteria checks: In [6], the checks of the pruning criteria of equations (6) and (7) have been simplified to a single pruning-criterion check of equation (7) in order to reduce hardware complexity, at the cost of a slight increase of N_en. For the SISO STS SD architecture proposed in this paper, the implementation of two different pruning criteria in unit ③ is mandatory to prevent a further significant increase of N_en. In order to avoid extra delays on the critical path, the pruning-criteria checks are not implemented as maximum searches but as pairs of $M_\mathrm{T}2^Q$ fully parallel comparators $\mathcal{M}_{\mathrm{prn},j}^{\mathrm{down}} > \lambda_{i,b}^{\overline{\mathrm{MAP}}}$ and $\mathcal{M}_{\mathrm{prn},j}^{\mathrm{sibl.}} > \lambda_{i,b}^{\overline{\mathrm{MAP}}}$, followed by simple bit-masking and combining.

V. ASIC Synthesis Results

The architecture presented in the previous section has been implemented in VHDL including parameters for word lengths, M_T, QAM order and a switch to enable/disable soft-input support. A representative set of parameter combinations has been instantiated by layout-aware gate-level synthesis[1].

Since both the soft-output-only base architecture and the SISO architecture follow the ONPC principle, their throughput Θ can be determined by

$$\Theta = \frac{rQM_\mathrm{T}}{\mathbb{E}[N_\mathrm{en}]} f_\mathrm{clk} \quad [bit/s] \tag{11}$$

with r being the code rate and $\mathbb{E}[N_\mathrm{en}]$ being the average N_en. The curves for the iterative Θ and the cumulative $\mathbb{E}[N_\mathrm{en}]$ for a 4×4 16-QAM MIMO system[2] achieving a frame error rate (FER) of 1 % are given in Figure 3, including as a reference the cumulative $\mathbb{E}[N_\mathrm{en}]$ obtained by SE ordering and floating-point operations. In the 4^th iteration the hybrid-enumeration algorithm introduces an overhead of less than 28 % in terms of $\mathbb{E}[N_\mathrm{en}]$. The least-effort throughput in Figure 3 is derived from equation (11) by selecting the minimum cumulative $\mathbb{E}[N_\mathrm{en}]$ among all iterations for a specific SNR. The intersections of the cumulative $\mathbb{E}[N_\mathrm{en}]$ curves determine the SNR points for changing the number of iterations. In Figure 3 the switching points are marked by ① (1 $\rightleftarrows$ 2 iterations), by ② (2 $\rightleftarrows$ 3 iterations) and by ③ (3 $\rightleftarrows$ 4 iterations).

Area and delay of this architecture are quite sensitive to the fixed-point word lengths. Therefore, the word lengths have

[1] UMC 90 nm standard-performance CMOS library, typical case, Synopsys Design Compiler 2009.06-sp1 in topographical mode.

[2] Throughout this paper we use a system with an i.i.d. Rayleigh fading channel, perfect channel knowledge and SQRD [10]. The BICM transmission is set up with a convolutional channel code (rate $1/2$, generator polynomials $[133_o, 171_o]$, constraint length 7) decoded by a max-log BCJR channel decoder with perfect termination knowledge and an S-random interleaver corresponding to 512 information bits. The SNR is defined as $\mathrm{SNR} = M_\mathrm{T}E_\mathrm{s}/N_0$, with $E_\mathrm{s} = \mathbb{E}[|s|^2], s \in \mathcal{O}$. $\mathrm{P}[s_i]$ is approximated by equation (9). The VLSI architecture internally operates on normalized metrics $\mathcal{M}_\mathrm{norm.} = N_0\mathcal{M}$ to avoid division by N_0, normalized clipping levels are given by $N_0 L_\mathrm{max}^\mathrm{E}$.

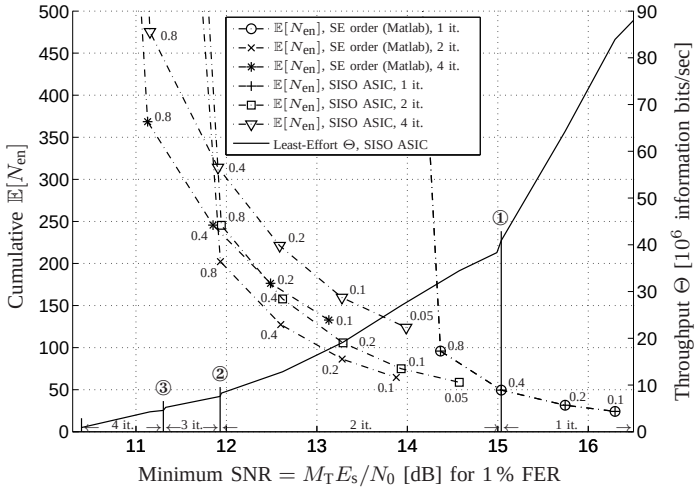


Fig. 3. Cumulative $\mathbb{E}[N_{\text{en}}]$ and iterative least-effort throughput Θ over minimum SNR for 1 % FER for the 4×4 16-QAM architecture. Numbers annotated to cumulative $\mathbb{E}[N_{\text{en}}]$ curves are normalized clipping levels $N_0 L^{\text{E}}_{\text{max}}$. As in [5], one iteration is defined as one use of the SISO MIMO demapper and the SISO channel decoder (1^{st} iteration corresponds to soft-output-only SD).

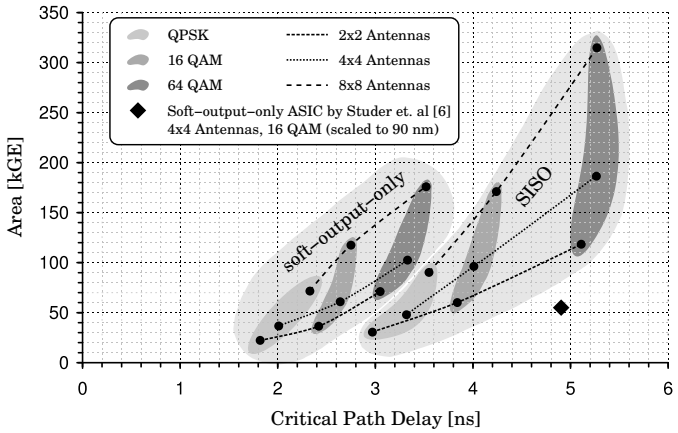


Fig. 4. Parametrization design space of the proposed STS SD architecture. Area is measured in gate equivalents (GEs). One GE corresponds to the area of a two-input drive-one NAND gate.

been carefully selected to make the FER-performance loss negligible with respect to floating-point operation[3].

Figure 4 shows the synthesis results for representative parameter sets. The results for the soft-output-only case are comparable to the implementation published in [6]. Since the two base architectures are similar, they are close in terms of area. The timing differs, mainly for two reasons. First, Figure 4 shows pre-layout synthesis results for a 90 nm technology whereas those in [6] are post-layout results for a 250 nm technology scaled to 90 nm by $f_{90} \approx \frac{250}{90} f_{250}$. Second, the architectures differ in their pipeline and enumeration schemes.

By enabling soft-input processing for the 4×4 16-QAM reference, the area increases by 57 % from 61 kGates to 96 kGates, while the clock frequency degrades by 34 % from 379 MHz to 250 MHz. We can conclude that the additional cost for soft-input is affordable at the prospect of working at lower SNR regimes with iterative systems.

The proposed architecture scales almost linearly with M_{T} in terms of area. The critical path degrades only by less than 10 % when doubling M_{T}. When increasing the QAM order by a factor of 4 in the soft-input case, the area is less than doubled while the frequency degrades by less than 20-25 %, despite the enumeration being significantly affected.

VI. Conclusion

To our best knowledge, we introduced the first SISO STS SD architecture, enabling iterative STS SD-based receivers. The parametrized architecture offers very good scalability over M_{T} and the QAM order. The approximate hybrid-enumeration method enables the implementation of iterative STS-based MIMO receivers, although high data-rate communication systems may require multiple parallel SD instances to meet the throughput constraints. We believe that the algorithms and hardware-design principles presented in this paper are suitable for most kinds of SD architectures. Our future development will focus on further enhancements of the architecture, based for instance on the ideas proposed in [6].

VII. Acknowledgement

The authors would like to thank Chun-Hao Liao, I-Wei Lai, Martin Senst, David Kammler, Andreas Minwegen, Uwe Deidersen, Konstantinos Nikitopoulos, Dan Zhang, Jeronimo Castrillon, Torsten Kempf, all reviewers and the editor for their valuable feedback and support.

References

[1] B. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, March 2003.

[2] S. Chen and T. Zhang, "Low power soft-output signal detector design for wireless MIMO communication systems," in *ISLPED '07: Proc. of the 2007 international symposium on low power electronics and design*. New York, NY, USA: ACM, August 2007, pp. 232–237.

[3] M. Li *et al.*, "Selective spanning with fast enumeration: A near maximum-likelihood MIMO detector designed for parallel programmable baseband architectures," in *Proc. IEEE International Conference on Communications ICC '08*, May 2008, pp. 737–741.

[4] S. Laraway and B. Farhang-Boroujeny, "Implementation of a markov chain monte carlo based multiuser/mimo detector," *IEEE Trans. Circuits Syst. I*, vol. 56, no. 1, pp. 246–255, January 2009.

[5] C. Studer and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," June 2009. http://arxiv.org/abs/0906.0840

[6] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: algorithms and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 290–300, February 2008.

[7] B. Mennenga and G. Fettweis, "Search sequence determination for tree search based detection algorithms," in *Proc. IEEE Sarnoff Symposium*, April 2009, pp. 1–6.

[8] C.-H. Liao *et al.*, "Combining orthogonalized partial metrics: Efficient enumeration for soft-input sphere decoder," in *Proc. IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, September 2009.

[9] C. P. Schnorr and M. Euchner, "Lattice basis reduction: improved practical algorithms and solving subset sum problems," *Math. Program.*, vol. 66, no. 2, pp. 181–199, August 1994.

[10] D. Wübben *et al.*, "Efficient algorithm for decoding layered space-time codes," *Electronics Letters*, vol. 37, no. 22, pp. 1348–1350, October 2001.

[11] C. Hess *et al.*, "Reduced-complexity MIMO detector with close-to ML error rate performance," in *Proc. of the 17th ACM Great Lakes Symposium on VLSI (GLSVLSI)*, March 2007, pp. 200–203.

[3] Word lengths [integer.fractional] for 4×4 16 QAM: $\tilde{y_i}[6.7]$, $R_{i,j}[4.7]$, $L^{\text{A}}_{i,b}[9.5]$, $L^{\text{E}}_{i,b}[9.5]$, $\mathcal{M}_{\{\text{C,A,P}\}}[9.6]$. A QAM-order increase of factor 4 requires one more integer bit for $\tilde{y_i}$ per real/imaginary part and two more integer bits for $\mathcal{M}_{\{\text{C,A,P}\}}$, $L^{\text{A}}_{i,b}$ and $L^{\text{E}}_{i,b}$. Doubling M_{T} requires one more integer bit for $\mathcal{M}_{\{\text{C,A,P}\}}$, $L^{\text{A}}_{i,b}$ and $L^{\text{E}}_{i,b}$.